一种针对机器阅读理解中答案获取的序列生成模型 *

霍　欢 [1,2]，邹依婷 [1]，金轩城 [1]，黄君扬 [1]，薛瑶环 [1]

(1. 上海理工大学 光电信息与计算机工程学院，上海 200093; 2. 复旦大学 上海市数据科学重点实验室，上海 201203)

摘　要：机器阅读理解中的答案获取是根据问题选择或者抽象释义出文章中的内容，但得到的序列容易出现表述不准确与信息冗余的问题。针对机器阅读理解任务中的答案获取提出一种序列生成模型 SGN。首先，SGN 在问题矩阵空间获取问题与文章的匹配表示，并参照潜在的问题信息，生成当前节点的词向量；然后，使用一个选择门结构从文章或者字典中选择当前词汇，并且自发学习和归纳 OOV（out-of-vocabulary）单词，解决语义表述不准确的问题。最后，使用改进的覆盖机制，消除生成序列中的冗余问题，从而提高可读性。实验通过人工数据集 SQuAD 进行验证，其结果表明，在阅读理解任务上 SGN 生成的目标序列与基准模型 Seq2Seq 相比可读性更加优异，并且与原文语义更贴近。

关键词：答案获取；序列模型；OOV；覆盖机制

中图分类号：TP391　　doi: 10.3969/j.issn.1001-3695.2018.08.0616

Sequence generation model for answer acquisition to machine reading comprehension

Huo Huan[1, 2], Zou Yiting[1], Jin Xuancheng[1], Huang Junyang[1], Xue Yaohuan[1]

(1. *School of Optical-Electrical & Computer Engineering, University of Shanghai for Science & Technology, Shanghai* 200093, *China*; 2. *Shanghai Key Laboratory of Data Science, Fudan University, Shanghai* 201203, *China*)

Abstract: Answer acquisition to Machine Reading Comprehension focuses on problem selection or abstract interpretation of the content of the article, but the sequence obtained is prone to problems of inaccurate representation and redundant information. A sequence generation model SGN is proposed for answer acquisition in the machine reading comprehension task. First, the SGN obtains the matching expression between problem and article in problem matrix space, and refers to the potential problem information to generate the word vector of the current node. Then, using a selection gate structure to select the current vocabulary from the article or dictionary, and spontaneously learns and generates OOV (Out-Of-Vocabulary) word to solve the problem of inaccurate semantic representation. Finally, use improved Coverage Mechanism to eliminates redundancies in the generated sequence and improve readability. The experiments adopt the artificial data set SQuAD. The results show that the target sequence generated by SGN is more readable than the benchmark model Seq2Seq and is closer to the original semantics.

Key words: answer acquisition; sequence generation model; OOV(out-of-vocabulary); coverage mechanism

0　引言

机器阅读理解任务是指让计算机阅读、理解一篇文章，然后针对与文中信息相关的问题给出答案。针对回答方式，可以分为填空型阅读理解与问答型阅读理解两种形式[1]。问答型阅读理解中以“what”“when”“how many”等疑问词提问。有的问答型阅读理解问题答案仅包含一个词汇，其处理过程与填空型阅读理解相类似，但对于复杂问题（如“why”“how”等疑问词开头的疑问句），需要结合上下文推理获取问题答案[2]。在问答型阅读理解任务上，机器给出答案的准确度与人的准确度相接近，但是机器生成问答型阅读理解答案时，依旧存在答案语义与问题句法间存在表述不一致的问题，如文章“With an estimated completion date of 2020, the Barack Obama Presidential Center will be housed at the university and include…”，问题“In what year will the Barack Obama Presidential Center be finished?”，机器给出的答案为“2020”。但从问题和目标答案的语义与结构可以看出，标准答案结构应该为“In 2020”，所以解决机器阅读理解任务中的答案生成问题，需要在阅读原文的基础上，理解问题的疑问词语义才能获取问题答案。

基于软注意力机制的 Seq2Seq 网络[3]能重新访问输入序列，并且注意力机制能动态选取输入序列中的列向量（即输入的每个词向量），从而对输入序列进行压缩。现有的基于 Seq2Seq 的答案获取技术可以分为两类，一类结合 Pointer Network 获取答案的起止索引[4~8]；另一类则是对输入的文章词向量进行概括，结合问题的语义信息，对多个相关或相似的词向量进行抽象生成，获得一个简要的序列[9~12]。使用第二种方法获得的答案序列，容易将原序列中多次出现的重要

收稿日期：2018-08-12；**修回日期：**2018-10-08　**基金项目：**国家自然科学基金资助项目（61003031）；上海重点科技攻关项目（14511107902）；上海市工程中心建设项目（GCZX14014）；上海市一流学科建设项目（XTKX2012）；上海市数据科学重点实验室开放课题资助课题（201609060003）；沪江基金研究基地专项项目（C14001）

作者简介：霍欢（1979-），女，副教授，博士，主要研究方向为数据管理、数据分析及数据挖掘；邹依婷（1994），女，硕士，主要研究方向为自然语言处理；金轩城（1995-），男，学士，主要研究方向为大数据；黄君扬（1995-），男，学士，主要研究方向为大数据；薛瑶环（1993-），女，硕士，主要研究方向为自然语言处理.

词向量重复输出，从而产生信息冗余。此外，大部分阅读理解模型都面临 OOV（out-of-vocabulary）问题，即模型解码预测目标答案时，Seq2Seq 模型不能解码出字典以外的词汇，从而得不到目标词向量[13]。

因此，在避免信息冗余的条件下，通过高度概括阅读理解的文章信息并准确表示问题答案，是个具有挑战性的任务。本文在 Seq2Seq 模型的基础上提出一种改进的答案—问题序列生成模型 SGN（sequence generation network）。针对答案获取问题，首先，SGN 使用改进的 Match-LSTM 网络替换 Seq2Seq 模型的编码层，改进的 Match-LSTM 能用文章在问题空间投影矩阵和原文章向量矩阵，获取包含问题与文章特征的匹配表示矩阵；其次，在匹配表示与预测当前节点相应的词向量前，设计一个选择门结构，一方面决定当前词汇来源，另一方面让模型自学习和归纳预测过程中产生的 OOV 词汇，以便模型能准确解码 OOV 词汇；最后，为了解决生成序列中产生信息冗余的问题，SGN 使用改进的覆盖（coverage）机制 [14]，让 SGN 模型可以在之前输出词向量的基础上解码当前节点词汇。

本文的贡献主要有如下三点：a) 提出了一种序列生成模型 SGN，在文章与问题匹配的基础上，利用节点生成的方式，解决生成序列中信息冗余和表述不准确的问题；b) SGN 使用一个选择门结构计算当前节点的词向量和选择概率，并利用选择概率选择基于匹配表示的词汇，对 OOV 单词进行学习并归纳到字典。因为每次仅生成一个词汇，所以能解决生成序列过程中产生的语义表述问题；c) SGN 模型使用了改进的覆盖机制，对 SGN 模型的解码层进行修正，能够在已经输出的词汇上解码当前词汇，从而消除生成序列中产生的冗余信息。

1　背景知识

1.1　问题定义

机器阅读理解中，通常用三元组 $<Q, D, A>$ 的形式描述将阅读理解任务，其中，$Q=(w_1, w_2, \dots, w_{|Q|})$ 表示可被回答的事实性问题；$D=(w_1, w_2, \dots, w_{|D|})$ 表示回答问题时参考的文章；$A=(w_1, w_2, \dots w_{|A|})$ 表示问题答案；$|Q|$、$|D|$ 和 $|A|$ 表示问题、文章以及答案中所包含的单词个数。问答型阅读理解需要对问题信息进行适当的推理，再对检索到的文章片段进行归纳、生成答案，即 $p(A|Q,D)$；然后，使用模型最终生成的序列与目标序列的匹配程度来评估模型生成序列的能力。

1.2　Match-LSTM 框架

Match-LSTM 是由 Wang 等人[15]提出的一个 LSTM（long-short term memory）[16]框架，用来处理文本蕴涵问题，旨在判断假设句的含义能否根据前提句推断而来。模型使用标准注意力机制，使用 Match-LSTM 逐字匹配词嵌入的前提与假设两个句子并进行分类。此外，LSTM 中的记忆单元能对错误匹配进行记忆。Match-LSTM 的模型如图 1 所示。

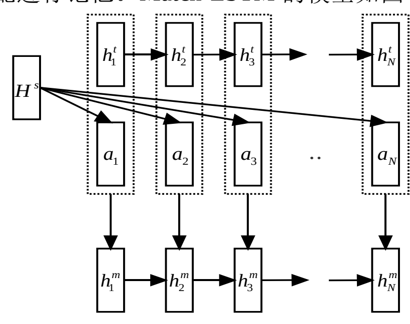


图 1　Match-LSTM 网络图

Fig. 1　Match-LSTM network

$$m_k = \begin{bmatrix} a_k \\ h_k^t \end{bmatrix} \tag{1}$$

$$i_k^m = \sigma(W^{mi} m_k + V^{mi} h_{k-1}^m + b^{mi}) \tag{2}$$

$$f_k^m = \sigma(W^{mf} m_k + V^{mf} h_{k-1}^m + b^{mf}) \tag{3}$$

$$o_k^m = \sigma(W^{mo} m_k + V^{mo} h_{k-1}^m + b^{mo}) \tag{4}$$

$$c_k^m = f_k^m \odot c_{k-1}^m + i_k^m \odot \tanh(W^{mc} m_k + V^{mc} h_{k-1}^m + b^{mc}) \tag{5}$$

$$h_k^m = o_k^m \odot \tanh(c_k^m) \tag{6}$$

其中：σ 是功能函数 sigmoid；$\odot$ 是两个向量元素间的点乘。此外，所有的权值矩阵参数 W 与 V 和权值向量参数 b 都可以在训练时学到。

2　SGN 模型

2.1　模型概览

图 2 为 SGN 模型概览，由词嵌入层、编码层与解码层构成。本节详细介绍 SGN 的基于 Match-LSTM 的编码层、节点单词生成，以及改进的覆盖机制，最后给出了 SGN 模型的目标函数。基于 Match-LSTM 的编码层获取文章中的每个词汇在问题空间的向量表示，节点单词生成是根据原文的上下文表示向量以及解码层隐含状态生成当前位置的词汇，改进的覆盖机制是用来解决生成序列中信息冗余问题。

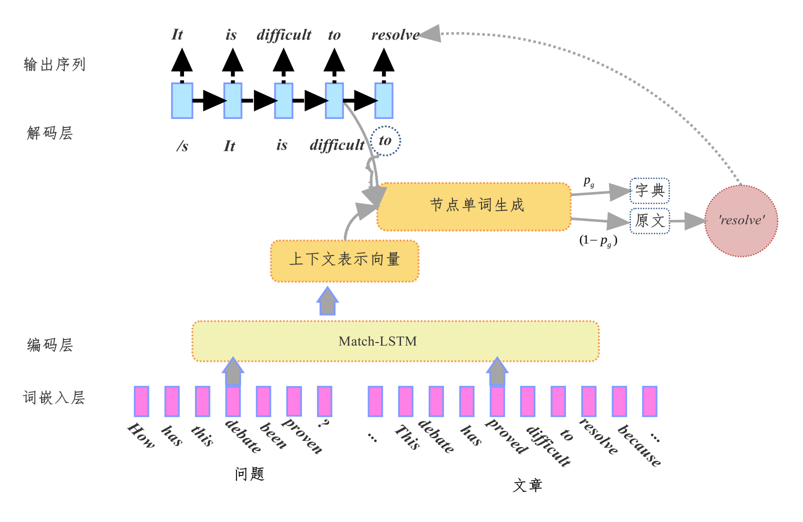


图 2　SGN 模型概览

Fig. 2　SGN model

2.2 建模

2.2.1 Match-LSTM 的编码层

基于 Match-LSTM 的编码层采用一个双向 *LSTM* 网络对词嵌入输出进行编码。采用双向 *LSTM* 网络能在隐含层保存每个词的前向状态以及后向状态，即最后输出的隐含层保存的是基于全文的状态。经过双向 *LSTM*(*BiLSTM*) 后，问题与文章的隐含层状态为

$$h_Q = BiLSTM(Q); h_D = BiLSTM(D) \tag{7}$$

其中：问题的隐含状态表示为 $h_Q \in \mathbb{R}^{l \times Q}$；文章的隐含状态表示为 $h_D \in \mathbb{R}^{l \times D}$ 。

为了提升模型的计算效率，本文简化了 Match-LSTM 网络设计的注意力机制，计算方式如式（8）所示。

$$A = soft\max((W_Q h_Q + b_Q \otimes e_Q) \bullet h_D) \tag{8}$$

其中：$W_Q \in \mathbb{R}^{l \times l}$ 、$b_Q \in \mathbb{R}^{l}$ 是模型学习所获得的参数；$b_Q \otimes e_Q$ 的结果为列向量 b_Q 重复 l 次形成一个 $l \times l$ 的矩阵，形成大小为 $l \times l$ 的矩阵。使用注意力矩阵 $A \in \mathbb{R}^{Q \times D}$ 获取文章中每个词汇在整个问题空间上的向量表示为

$$\bar{h}_D = h_Q \bullet A \tag{9}$$

其中：$\bar{h}_D \in \mathbb{R}^{l \times D}$ 。此时文章中的每个词都由整个问题表示，即 $\bar{h}_D$ 中每个行向量为文章中的词都是在整个问题空间上的向量表示。为了获取投影后的文章表示与原始文章间的联合匹配表示。首先，模型计算原始 h_D 与 $\bar{h}_D$ 间的特征向量 $Z(h_D, \bar{h}_D)$，获取 h_D 与 $\bar{h}_D$ 间的相似性；然后，通过一个单层前向 *LSTM*，为特征打分，从而得到两者的匹配度 M 。数学过程如下：

$$Z(h_D, \bar{h}_D) = [h_D, \bar{h}_D, h_D \circ \bar{h}_D, |h_D - \bar{h}_D|, h_D W \bar{h}_D] \tag{10}$$

$$M = LSTM(Z(h_D, \bar{h}_D)) \tag{11}$$

其中：W 是可以通过模型训练获得；“∘”表示矩阵中每个元素点乘。最后，为了表示一篇文章与问题词汇的匹配表示，模型添加一个双向 LSTM 聚合所有的词匹配表示，即

$$H = BiLSTM(M) \tag{12}$$

此时，文章中第 t 个词汇的注意力为

$$\alpha^t = \text{soft}\max(H^t) \tag{13}$$

第 t 个词汇 w_t 的上下文向量，用当前隐含层状态与注意力的加权平来表示：

$$h_t = \sum_i \alpha_i^t H^t \tag{14}$$

2.2.2 节点单词生成

本文的 SGN 模型是基于 Seq2Seq 和 Pointer Network[17]，因为模型既可以从原文中复制原始词汇，也可以用问题与原文的隐含特征，从固定的字典中概括出来。在时刻 t，词汇生成的概率 $p_g \in [0,1]$，即当 p_g=1 时，模型需要从字典中概括出词汇 e_t；当 p_g=0，则直接从文章词汇中进行复制，其功能表达式如式（15）所示。

$$p_g = sigmoid(W_h h_t + W_d d_t + W_S S_t + b) \tag{15}$$

其中：h_t 为当前时刻 t 上的上下文向量；d_t 为 Seq2Seq 解码在时刻 t 隐含层的状态；S_t 解码层的输入；b 为偏差。这些参数都可通过模型训练、学习而获得。选择文章中词汇或者生成单词表中的词汇，SGN 添加了一个门选择操作，用来决定单词源：

$$p(w_t = w) = p_g p_v + (1 - p_g) a_w \tag{16}$$

$$a_w = \sum_{w_i = w} \sum_{q=1}^{Q} A_{w_{qi}} \tag{17}$$

$$p_v = sigmoid(V'(Vd_t + Wh_t + b) + b') \tag{18}$$

通过这个门操作，模型能准确地选择单词词汇，并将其输出。当 p_g=1 或者 $\sum_{w_i=w}\sum_{q=1}^{Q} A_{w_{qi}}$=0 时，生成序列中第 t 个位置的单词用于文章、问题与字典为条件预测的结果进行输出；当 $p(w_t = w)$=0 时，则 w 为 OOV 词汇，将其上下文表示向量作为其在字典中的向量值并将其存放到字典中。

2.2.3 SGN 模型修正

重复是 Seq2Seq 模型的一个通病，即同义词的多次出现，这种现象在含复合句的长文本中的表现尤其明显。为此，SGN 模型改进了覆盖机制来修正原来 SGN 模型，通过使用一个覆盖向量 c_t，c_t 通过对前面的解码输出序列的注意力分布进行求和：

$$c_t = \sum_{t'=0}^{t-1} \alpha_d^{t'} \tag{19}$$

$$e_t = LSTM(W_h h_t + W_S S_t + W_c c_t + b) \tag{20}$$

其中：W_c 能通过模型训练、学到的参数；$\alpha_d^{t'}$ 表示源数据的注意力分布，即解码层输出的概率分布；c_t 表示的概率中包含到目前为止产生的词汇。显然 c_0=0，第一个解码单元的输入不为输出的词汇。

通过改进了覆盖机制，SGN 模型可以保证现在注意力机制所做的决定，即从原文复制或概括后从词典中复制过来，都参考了之前所做决定，并且能有效避免同一个单词或相似单词的重复出现，从而解决输出结果中产生的重复问题。

2.2.4 目标函数

SGN 模型采用有监督的方式进行训练。训练的输入特征为阅读理解的文章 D 与问题 Q，模型的输出为预测的问题答案 A，目标为数据集中给定的答案。生成序列中，模型在注意力覆盖范围内容易产生信息冗余本文使用注意力与覆盖的最小值表示覆盖机制产生的损失。此外，即便阅读理解答案能直接从原文以及字典中进行复制，或者自行生成，结果间依旧存在差异。目标函数中节点生成产生的代价为总代价的平均值。所以目标函数定义如下：

$$J = \log p(y^* | Q, D) + \frac{1}{T} \sum_{t=1:T} \log p(w_t) + \sum_t \min(\alpha_t, c_t) \tag{21}$$

3 实验

首先介绍实验所采用的数据集——斯坦福阅读理解数据集（SQuAD[18]）；然后说明模型参数的设置；最后对比分析效果最佳模型与实验采用的基准模型的实验结果。

3.1 数据集

SQuAD 数据集包含 536 篇文档，以及超过 1 000 000 个由专业人士根据文档信息提出的问题。提问者是根据自己对文档的理解与认知，提出相对应的问题，而不是直接从文档中提取短语或句子作为问题。此外，提问者还给出了对应问题的答案。答案是用文档中词、短语或者句子片段等构成的变长序列。SQuAD 数据集中，训练集包含 87 599 个实例，验证集包含 10 570 个实例，测试集数据则未公开。实验部分将使用公开两部分数据集（将近整个数据集 90%）来验证模型。然后将已获得数据集进行随机划分为训练集、验证集、测试集，所占已获得数据集的百分比分别为 80%、10%和 10%。

模型 SGN 需要归纳文章，然后对齐问题给出最终答案，SQuAD 数据集仅提供答案片段。为此，实验前需要对数据进

行预处理，将数据集中的答案根据问题结构获取答案对齐问题的标准化形式。答案标准化流程为：使用人工定义规则；然后使用 HMM 标注数据获取答案的主体部分；最后用依赖树[19]对主体词分词，并用答案替换主体词树型结构中的疑问词部分，得到标准答案。为了获取标准化的准确度，使用公开两部分数据集中 90%的数据训练机器标注、10%测试标注的准确度，并且从测试的最终结果中随机抽取 1 000 条数据进行人工评测。标注实验的准确率为 96. 231%，其中产生错误的因素包括 OOV 词汇、原文标注不准确，以及错误的语义分词。本实验最终采用的数据集是在标准化的实验结果上再次进行人工标注数据。其中，数据集的部分结果如表 1 所示。

表 1　标注后的部分案例

Table 1　Example cases after labeling

文章：With an estimated completion date of 2020, the Barack Obama Presidential Center will be housed at the university and include…
问题：In what year will the Barack Obama Presidential Center be finished?
标注后答案：In 2020
文章：the extinction of the dinosaurs and the wetter climate may have allowed the tropical rainforest to spread out across the continent.
问题：Which type of climate may have allowed the rainforest to spread across the continent?
标注后答案：the wetter climate
文章：Around the world many governments operate teacher's colleges, which are generally established to serve and protect the public interest through certifying
问题：Why would a teacher's college exist?
标注后答案：To serve and protect the public interest

其中文章中的方框为 SQuAD 数据集给出的标准答案。

本实验通过采用 SQuAD 答案标准化后的数据集中包含的 $<Q,D,A>$ 元组对训练 SGN 模型的参数，并验证模型的合理性与准确性。

3.2　实验设置

电脑配置 Intel/Xeon E5-2683V3 14 核 28 线程，NVIDIA GTX 1080 显卡，32 GB 内存，并使用 Tensorflow 深度学习框架实现。

1) 参数设置　SGN 模型的编码与解码隐含层使用的 LSTM Cell 个数为 600。词向量则在训练过程中，让模型抓取相应的词向量，本文使用的字典为 300 维预训练词向量表 840B-GloVe[20]。小批量训练 $\langle Q,D,A\rangle$ 对的大小为 32。当一个批训练中序列长度不一致时，选取中间长度最大的值，并且长度不足的序列后面填补空白向量，其向量值是一个 300 维的 0 向量。训练过程中迭代数为 60，当在验证集上连续 3 次迭代后的模型的准确度没有得到提升，甚至出现降低，则提前停止训练。实验保存模型在验证集上代价最低的一组模型参数，该参数作为最优模型在测试集上进行测试。

2) 超参数设置　使用 Adam[21]算法对模型参数进行优化。其中，第一动量系数 beta1 和第二动量系数 beta2 分别设为 0.9 和 0.999。初始学习率设为 0.001，epsilon 设置为 10E−8。为了加速梯度下降过程，实验时为全局每 1 000 训练步长设置了大小为 0.9 的衰减率。同时，为了防止在 SQuAD 数据集上测试 SGN 模型产生过拟合，训练模型参数时添加了 dropout 机制[22]，在模型的输入端和输出端、控制层的输出端随机关闭隐含层中 15%的神经元。此外，词嵌入层的嵌入权值是随机初始化并且服从区间[-0.05，0.05]上的均匀分布。网络中的隐藏层的使用的参数为一个随机初始化的正交矩阵。本文给出的答案生成的实验结果是在重新划分的数据集上获取的结果。对于 OOV 问题，SGN 模型没有预先对词嵌入层进行训练，而是在训练的过程中让模型自行学习、获得当前抓取词汇的向量表达。

实验首先实现了 SGN，即基于 Match-LSTM 的 SGN 网络；然后在 SGN 上扩展了改进的覆盖机制，记为 SGN+Coverage，覆盖机制使用式（21）的代价函数；为了证明 SGN+Coverage 匹配层的作用，实现了去除匹配层的 SGN，即用 600 神经元的双层 LSTM 网络替换 Match-LSTM 匹配层，再依据概率分布抽取局部权值最大的词汇。

实验采用的基准模型为 Seq2Seq+attention 模型，其中模型的参数由 SQuAD 答案标准化后的数据集的训练集进行训练，最优模型的参数选择则用验证集上损失最低的一组参数值。SGN 选择生成文本操作与 Gu 等人[23]设计的 CopyNet 类似。为了验证两者的性能，实验首先使用 CopyNet 预先定义的参数设置，使用 SQuAD 答案标准化后的数据集训练模型。 Seq2Seq+attention 与 CopyNet 都

表 2　测试集上获得的 ROUGE 值与 EM 值

Table 2　ROUGE and EM from test set

	测试集			
	ROUGE-1	ROUGE-2	ROUGE-L	EM(Exact Match)
Seq2Seq+attention（baseline）	53.87	32.82	50.23	33.52
CopyNet	56.81	36.08	53.58	35.79
SGN	57.49	36.27	53.66	36.61
SGN+Coverage	58.43	37.09	54.27	37.13
去除匹配层的 SGN	56.7	36.21	53.41	36.02

添加对问题与文章进行编码的编码层，并且在编码层共享一个权值矩阵。字典外(OOV)的词汇进行词嵌入后，会被映射到预先定义的 UNK 标记。

3.3　结果分析

3.3.1 准确度分析

SQuAD 提供了模型估计脚本，本实验时使用 ROUGE（recall-oriented understudy for gisting evaluation）[24]和准确匹配（exact match，EM）对模型的输出结果进行估计，实验结果如表 2 所示。其中，获取自动文摘指标 ROUGE 包括 ROUGE-N，$(N=[1,2])$ 和 ROUGE-L。自动生成的摘要或翻译与一组参考摘要（通常是人工生成的）进行比较计算，得出相应的分值，用来衡量自动生成的摘要或翻译与参考摘要之间的"相似度"。其计算公式为

$$ROUGE-N=\frac{\sum_{S\in(\text{参考摘要集})}\sum_{gram_n\in S}count_{match}(gram_n)}{\sum_{S\in(\text{参考摘要集})}\sum_{gram_n\in S}count(gram_n)} \tag{22}$$

其中：N 为元词个数。

ROUGE-L 的计算公式为

$$R_{lcs}=\frac{1}{m}LCS(X,Y) \tag{23}$$

$$P_{lcs}=\frac{1}{n}LCS(X,Y) \tag{24}$$

$$F_{lcs}=(1+\beta^2)R_{lcs}P_{lcs}/(R_{lcs}+\beta^2P_{lcs}) \tag{25}$$

其中：X 为参考摘要；长度为 m；Y 为候选摘要；长度为 n；用准确率 P_{lcs} 值来衡量摘要 X 与 Y 的相似度，评测过程中 $\beta\to\infty$，所以只考虑召回率 R_{lcs}。EM 值是生成结果与目标答案完全匹配的奖励。

从表 1 可以看出，在召回率评估和准确匹配上，准确模

型在 ROUGE 和 EM 上的表现并不是很理想。而 SGN 模型与 SGN+Coverage 模型取得了一个比较高的分值，同时，SGN 与 SGN+Coverage 的输出序列与目标结果间的匹配值也明显高于基准模型。其中，表现最好的 SGN+Coverage 提升的分值为+4.56 ROUGE-L、+ 4.27 ROUGE-2、+4.04ROUGE-L、+3.61EM。去除匹配层后问题与文章只能依赖词间的相关度抽取最相关的词汇，其分值相对于基准模型有所提升，但同 CopyNet 来比，在准确度上仅提升了 0.23。SGN 与 CopyNet 相比，两者结构上相似，得到的 ROUGE 与 EM 值也非常接近。而 SGN+Coverage 的分值同 CopyNet 比较，提升了 +1.62 ROUGE-1、+1.01 ROUGE-2、+0.69 ROUGE-L、+1.34 EM。SQuAD 数据集的答案为文章中的词、短语以及句子或句子片段，所以 CopyNet 与 SGN 对源数据推理后所获得实验结果基本相似，获得的实验结果很接近。但使用改进的覆盖机制优化 SGN 模型后，模型能在已经预测出的词汇上进行判断，并决定当前位置应该输出的词汇，从而获得较准确的答案信息。所以有理由相信 SGN+Coverage 可以获得较高的 ROUGE 值，一部分与 EM 值与数据集的特征分不开，另一部分取决于 SGN 的匹配层，匹配层使得文章数据更加依赖于潜在的问题信息，使结果与最终目标序列更加匹配。

3.3.2 生成序列重复率分析

图 3 为 SGN 和 SGN+Coverage 在测试集中获取的输出序列中 n 元词（gram-n，n=1,2,3,4），以及最长公共子序列中的重复率。由 SGN 和 SGN+Coverage 两者的输出重复率比可以看出，使用改进的覆盖机制的 SGN 网络，能有效减少生成序列中单词的重复个数，并且其生成序列中的 n 元词的重复数与目标序列接近。虽然在训练参数时，对覆盖机制的训练次数少（所有训练次数的 0.8%左右），但在重复产生信息冗余的问题上基本上得到消除。

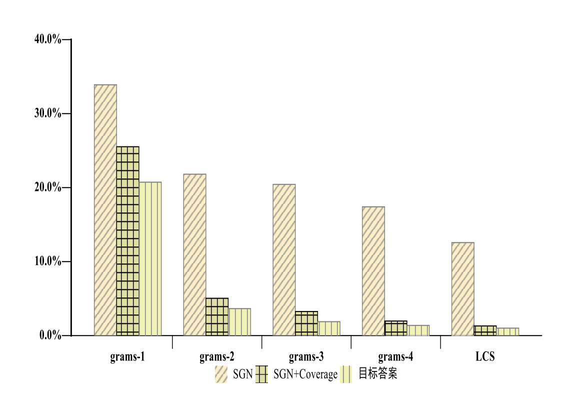
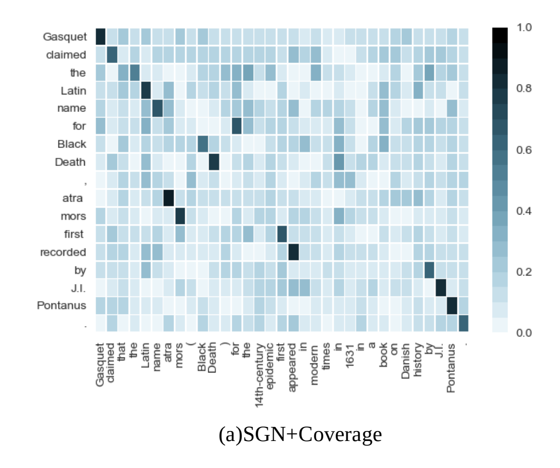


图 3 输出序列与目标序列中重复率对比

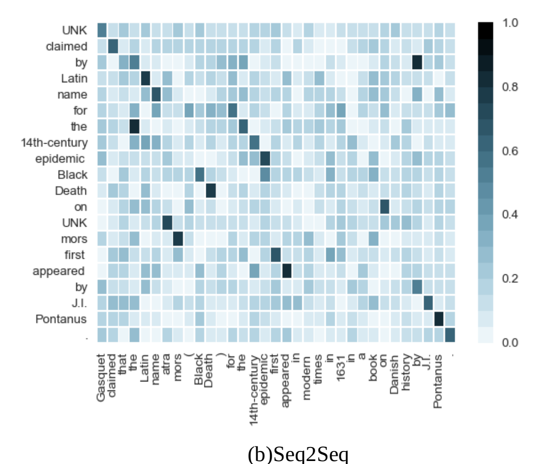
Fig. 3 Repetition rate comparison among sequences

3.3.3 案例分析

图 4 为 SGN+Coverage 与基准模型 Seq2Seq 在验证集上部分文章词汇与归纳序列中词汇的热力图。其中颜色深浅表示词间的相关度，颜色越深表示其相关程度越高。从最终生成的序列来看，Seq2Seq 获得的序列存在语法问题（“claimed by”）以及事实细节的描述错误（“appeared by J.I.”，原序列为“appeared in a book”），容易用原序列中的词汇替代事实性描述。Seq2Seq 词嵌入的词汇是参考已经训练好的字典，对于字典中不存在的词汇（如“Gasquet”“atra”）使用 UNK 标记，最终得到的序列结果可读性并不高。SGN+Coverage 模型自身具备对新词汇进行学习的能力，这在一定程度上解决了 OOV 的问题，提升了 OOV 词汇的召回率，并且最终获得结果具备一定的语法。此外，对原序列中的“appeared”生成中生成了具备被动语态的“recorded”，从整体来看符合原序列的描述。虽然生成过程中原序列词向量与目标序列词向量间的相关度过于随机，但从图 4 可以看出，基于 Seq2Seq 的 SGN+Coverage 模型除了动态选择原输入词向量，缩减了输入序列的长度，同时还通过适当地合并与替换原词向量，使得输出序列的细节描述更加准确。



(a)SGN+Coverage



(b)Seq2Seq

图 4 生成序列的热力图对比

Fig. 4 Heat map comparison between generated sequences

4 相关工作

阅读理解对机器来说是个巨大挑战，它要求机器掌握人类语言以及各种知识。最近，已经有许多神经网络模型在 SQuAD[18]数据集上采用抽取的方式进行机器阅读理解验证。Dynamic Chunk Reader [1]是 IBM Watson 团队提出，是通过抽取原文中的答案候选集，然后对候选集进行排序。RASOR[4]介绍了一种使用文章的独立表示和文章对齐问题表示的结构抽取候选答案，并且为候选答案打分，模型的最终输出为分值最高的候选答案。Zhang 等人[5]是在神经网络的基础上，通过引入句法信息来理解阅读理解中的问题，使用 TreeLSTM 捕获文章中远距离的迭代信息，并且对不同类型的问题单独建模。Wang 等人[6]首次在 SQuAD 数据集上使用端对端的神经网络进行测试，通过结合文章与问题，并采用 pointer Network 来决定答案片段所在位置。High Maxout Network [7]一个动态解码的神经网络，用来提升解码效率。BIDAF [8]是通过使用一个双向注意力获取问题注意表示的上下文。

生成任务需要对输入序列进行抽象释义，主要工作包括 CopyNet[23]，是在 Seq2Seq + Attention 的基础上，引入了拷贝机制，“复制”原序列中的重要信息并决定“粘贴”位置。Du

等人[12]利用基本 attention 的问题生成模型，模型依据输入的句子(段落)生成相应的问题，从而解决阅读理解任务。序列对序列学习模型 COREQA[25]是通过编码—解码框架结合复制（copying）和索引（retrieve）两个机制，实现答案生成。GenQA[26]只能在 Seq2Seq 模型上处理单个事实的简单问题。Miao 等人[27]在 Seq2Seq 模型上使用一个用于推理的变分自动编码器，解决阅读理解任务，其生成模型首先从语言模型中获取潜在的句子概括，然后根据潜在的总结得到最终的结果。

但是 SGN 为了解决阅读理解任务，使用了一个简单的门结构实现原文生成或者从字典中复制两个操作。此外，为了获取原文向量中更多的潜在特征，SGN 通过在问题空间的文章向量表示与文章原始向量表示，获取两者的匹配向量。为了解决同词或同义词重复出现的问题，本文采用了覆盖机制修正 Seq2Seq 学习模型，使得最后实验的结果更加准确。

5　结束语

本文提出了一种序列生成模型 SGN，在文章与问题匹配的基础上，利用节点生成的方式，解决生成序列中信息冗余和表述不准确的问题。SGN 使用一个选择门结构计算当前节点的词向量和选择概率，并利用选择概率选择基于匹配表示的词汇，对 OOV 单词进行学习并归纳到字典。因为每次仅生成一个词汇，所以能解决生成序列过程中产生的语义表述问题。SGN 模型使用改进了的覆盖机制，对 SGN 模型的解码层进行修正，能够在已经输出了的词汇上解码当前词汇，从而消除生成序列中产生的冗余信息。通过 SGN 在答案生成上的测试结果可知，本文提出的 SGN 和已有模型在保持生成序列的准确匹配的同时，获得的生成序列在细节描述上也比较准确。在未来的工作中，将对模型中的选择门结构进行优化，以便适用于更高阶阅读理解生成任务，如开放域的阅读理解任务等。

参考文献：

[1] Yu Yang, Zhang Wei, Hasan K, *et al.* End-to-end reading comprehension with dynamic answer chunk ranking [EB/OL]. (2016) [2018-09-30]. https://arxiv. org/abs/1610. 09996.

[2] Choi E, Hewlett D, Uszkoreit J, *et al.* Coarse-to-fine question answering for long documents [C]// Proc of the 55th Annual Meeting of the Association for Computational Linguistics. Vancouver: The Association for Computational Linguistics, 2017: 209-220.

[3] Bahdanau D, Cho K, Bengio Y. Neural machine translation by jointly learning to align and translate [EB/OL]. (2014) [2018-09-30]. http://arxiv. org/abs/1409. 0473.

[4] Lee K, Salant S, Kwiatkowski T, *et al.* Learning recurrent span representations for extractive question answering [EB/OL]. (2016) [2018-09-30]. http://arxiv. org/abs/1611. 01436.

[5] Zhang Junbei, Zhu Xiaodan, Chen Qian, *et al.* Exploring question understanding and adaptation in neural-network-based question answering [EB/OL]. (2017) [2018-09-30]. http://arxiv. org/abs/1703. 04617.

[6] Wang Shuohang, Jiang Jing. Machine comprehension using match-lstm and answer pointer [EB/OL]. (2016) [2018-09-30]. http://arxiv. org/abs/1608. 07905.

[7] Xiong Caiming, Zhong V, Socher R. Dynamic coattention networks for question answering [EB/OL]. (2016) [2018-09-30]. http://arxiv. org/abs/1611. 01604.

[8] Seo M, Kembhavi A, Farhadi A, *et al.* Bidirectional attention flow for machine comprehension [EB/OL]. (2016) [2018-09-30]. http://arxiv. org/abs/1611. 01603.

[9] Nallapati R, Zhai Feifei, Zhou Bowen. SummaRuNNer: a recurrent neural network based sequence model for extractive summarization of documents [C]// Proc of the 31st Conference on Artificial Intelligence. San Francisco: AAAI Press, 2017: 3075-3081.

[10] Cao Ziqiang, Luo Chuwei, Li Wenjie, *et al.* Joint copying and restricted generation for paraphrase [C]// Proc of the 31st Association for the Advancement of Artificial Intelligence. San Francisco: AAAI Press, 2016: 3152-3158.

[11] Nallapati R, Zhou Bowen, Santos C N D, *et al.* Abstractive text summarization using sequence-to-sequence rnns and beyond [C]// Proc of the 20th Conference on Computational Natural Language Learning. Berlin: Association for Computational Linguistics, 2016: 280-290.

[12] Du Xinya, Shao Junru, Cardie C. Learning to ask: neural question generation for reading comprehension [C]// Proc of the 55th Annual Meeting of the Association for Computational Linguistics. Vancouver: Association for Computational Linguistics, 2017: 1342-1352.

[13] See A, Liu Peter J, Manning C D. Get to the point: summarization with pointer-generator networks [C]// Proc of the 55th Annual Meeting of the Association for Computational Linguistics. Vancouver: Association for Computational Linguistics, 2017: 1073-1083.

[14] Tu Zhaopeng, Lu Zhengdong, Liu Yang, *et al.* Modeling coverage for translation [C]// Proc of the 54th Annual Meeting of the Association for Computational Linguistics. Berlin: Association for Computer Linguistic, 2016: 76-85.

[15] Wang Shuohang, Jiang Jing. Learning natural language inference with LSTM [C]// Proc of the 16th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies. San Diego: Association for Computational Linguistics, 2015: 1442-1451.

[16] Hochreiter S, Schmidhuber J. Long short-term memory [J]. Neural Computation, 1997, 9 (8): 1735-1780.

[17] Vinyals O, Fortunato M, Jaitly N. Pointer networks [C]// Proc of the 29th Advances in Neural Information Processing Systems. 2015: 2692-2700.

[18] Rajpurkar P, Zhang Jian, Lopyrev K, *et al.* [C]// Proc of the 21th Conference on Empirical Methods in Natural Language Processing. Austin: Association for Computational Linguistics. 2016: 2383-2392.

[19] Reddy S, Täckström O, Petrov S, *et al.* Universal semantic parsing [C]// Proc of the 22th Empirical Methods in Natural Language Processing. Copenhagen: Association for Computational Linguistics, 2017: 89-101.

[20] Pennington J, Socher R, Manning C. Glove: global vectors for word representation [C]// Proc of the 19th Empirical Methods in Natural Language Processing. Doha: Association for Computational Linguistics, 2014: 1532-1543.

[21] Kingma D P, Ba J. Adam: a method for stochastic optimization [EB/OL]. (2014) [2018-09-30]. http://arxiv. org/abs/1412. 6980.

[22] Srivastava N, Hinton G E, Krizhevsky A, *et al.* Dropout: a simple way to prevent neural networks from overfitting [J]. Journal of Machine Learning Research, 2014, 15 (1): 1929-1958.

[23] Gu Jiatao, Lu Zhengdong, Li Hang, *et al.* Incorporating copying mechanism in sequence-to-sequence learning [C]// Proc of the 54th Annual Meeting of the Association for Computational Linguistics. Berlin: Association for Computer Linguistics, 2016: 1631-1640.

[24] Flick C. ROUGE: a package for automatic evaluation of summaries

[C]// Proc of Workshop on Text Summarization Branches Out. 2004: 10.

[25] He Shizhu, Liu Cao, Liu Kang, *et al*. Generating natural answers by incorporating copying and retrieving mechanisms in sequence-to-sequence learning [C]// Proc of the 55th Annual Meeting of the Association for Computational Linguistics. Vancouver: Association for Computational Linguistics, 2017: 199-208.

[26] Yin Jun, Jiang Xin, Lu Zhengdong, *et al*. Neural generative question answering [C]// Proc of the 25th International Joint Conference on Artificial Intelligence. New York: IJCAI//AAAI Press, 2016: 2972-2978.

[27] Miao Yishu, Blunsom P. Language as a latent variable: discrete generative models for sentence compression [C]// Proc of the 21th Conference on Empirical Methods in Natural Language Processing. Austin: The Association for Computational Linguistics, 2016: 319-328.